UNIVERSITY OF WISCONSIN-MADISON
Computer Sciences Department

Operating Systems Depth Exam **Fall 2000**

Instructions: There are *seven* questions on this exam; answer *any six* of the following seven questions. This exam has three (3) pages.

Question 1. Network Address Translation:

IP addresses are becoming a scarce resource, being only 32 bits in length. With the rapidly growing Internet, this is becoming a serious problem. Organizations (even large ones) can reduce the number of IP addresses they need by using Network Address Translation (NAT). With NAT, an organization potentially needs only a few public IP addresses. NAT allows your local networks to have a separate private IP name space (using any addresses that you want), mapping a public Internet (IP, port number) pair onto a local private (IP, port number).

NAT usually runs on the computer or router (firewall) that sits between your local networks and the Internet.

A. Assume that a process on your local network wants to connect to a server (host address 128.105.167.44, port number 25) on the Internet. Explain how this connection would be made, illustrating how the sockets (IP addresses, port numbers) are mapped.
B. Assume that you have a server (such as a Web server on port 80) running somewhere on your local network. This Web server should be publicly available to users on the Internet. Explain how a process somewhere on the Internet would be able to connect to your server. What additional functionality is needed to support making connections to your server?
C. Suppose your local network has a Web server and a mail server, both visible to the Internet, with each server running on a different host on your local network. How would you handle this case?

Question 2. Remote Execution Security:

In the Condor system, I can run my programs remotely on another computer. My program is linked with a special Condor version of the C library; when my program runs, all system calls (such as read() or write()) made the program are sent back my home (desktop) machine and executed there.

A. We want to prevent my program from abusing the remote computer. Describe three things that my program might try to do that would violate the security of the remote computer, and describe a strategy for preventing this abuse.
B. We also want to prevent the owner of the remote machine (who has root/superuser privileges) from using my remote program as an inappropriate path to access the files or resources on my desktop computer. Describe one way in which the owner could make such an access, and describe

how you might prevent such an access.

Question 3. Distributed Shared Memory:

Assume that you implementing a page-level distributed shared memory (DSM) system, without having special operating system support. You can assume that you're building this system on top of your favorite Unix or NT platform.

A. Describe the basic components of a distributed shared memory system.
B. How can you detect a page fault? E.g., how can you detect an access to a page that is residing on another host?
C. What is sequential consistency? Why is this undesirable in a DSM system?

Question 4. Network-Attached Storage Devices:

Network-attached storage devices are a new storage paradigm. In this environment, disks each have some local processing capability and a network interface, thus enabling the addition of each disk directly to the network. Assume you are putting together a storage system that consists of hundreds of these devices, for a workload that consists of mostly small reads and writes (perhaps a database transactional workload).

A. Compare the performance, reliability, manageability, and security properties of network-attached storage with the old style of attaching disks to a server.
B. What RAID scheme would you use to organize the drives? Why?
C. Over time, decision support queries (mostly large, sequential reads) are also run over the database stored on your disks, and come to dominate both in terms of total traffic and as time. How might this affect your choice of RAID scheme? Assuming that you decide to switch schemes, how could this be accomplished on-line without losing data availability?

Question 5. OS Support for Databases:

In 1981, Michael Stonebraker wrote a paper entitled "Operating System Support for Database Management", where he discusses the many deficiencies of operating systems in support of database applications. Since then, however, operating systems have evolved substantially - or have they? In this question, we examine just how much they have evolved.

A. The first and foremost problem is buffer management. One particular problem occurs when performing (repeated) sequential scans of a large file. Describe the potential problem that occurs in the file cache, assuming a traditional LRU management scheme that is common in OS's.
B. What kind of mechanisms or policies or both would an operating system need to circumvent this problem? Give two possible solutions - you can use examples from real modern operating systems if you like.
C. Stonebraker also states that "although we have never been offered a good explanation of why

messages are so expensive, the fact remains that in most operating systems the cost for a round-trip message is several thousand instructions. For example, on the PDP-11/70 UNIX, the number is about 5,000." Why are messaging costs typically high? What can an OS designer do about it?

Question 6. A New Storage Technology:

A new device technology known as *MEMS* (or micro-electro-mechanical systems) is being used to build new types of storage devices. One promising MEMS-based storage design is described as follows: data is stored on a two-dimensional grid, accessible in blocks. Each block thus is addressed with a simple (x,y) tuple. When accessed, a "media sled" is electronically moved to the correct (x,y) location, and the desired block is read or written. Because the sled has mass, it must accelerate and decelerate when moving from one location to another.

A. If you had control over the hardware, what block size would you pick and why?
B. Assume that you are going to build a specialized file system for a web server on top of this device. How would you layout data and meta-data? (write down assumptions that you make)

Question 7. CPU Scheduling:

Lottery scheduling is a simple, randomized proportional-share scheduling technique. In this scheme, each job has some number of tickets, and at each quantum, a lottery is held, the winner of which runs that quantum. The more tickets a job has, the more likely it is to win (e.g., if job A has two tickets, and job B has 1, a random number between 0 and 2 inclusive will be chosen. If 0 or 1 is chosen, A will run, and if 2 is chosen, B will run).

A. You are hired to re-design lottery scheduling to remove the randomized aspect. Sketch out an algorithm that also provides "proportional sharing", but does so deterministically.
B How are the properties of each algorithm different, in terms of fairness and performance? Is one better than the other? How do both compare to standard multi-level feedback scheduling, again in terms of both fairness and performance?